

REMARKS

The present remarks supplement the remarks set forth in the Amendment and Response dated May 2, 2006. In the concluding remarks made on page 16 of such Amendment and Response, it was respectfully submitted that the discussion of '427 therein clearly indicates that (1) the applications on the hosting servers are separate from the replication service, (2) the replication servers do not act in response to application-specific policies of the application objects hosted on the hosting servers, (3) the structure of the replication servers is not part of the applications hosted on the hosting servers, and (4) no control function of the replication service is part of any application hosted by the hosting servers.

However, those remarks did not expressly refer to the Abstract of '427 or to the cited C12, L15-30. To avoid any misunderstandings with respect to '427, the following expressly refers to the Abstract of '427 and to the cited C12, L15-30. It is noted that on Office Action page 2, last 2 lines (re claim 1) reference was made to "(Abstract)" as support for teaching a system for providing application-specific strategies to a platform; and on Office Action page 4, re claim 8, the '427 Abstract was cited as teaching a method for starting an application having application-specific strategies. No specific relation to '427 or 8213 was made, but the discussion was of '427, thus the '427 Abstract is referenced here.

It is respectfully submitted that the '427 Abstract does not teach the claimed system for providing application-specific strategies to a platform; nor the claimed method for starting a child application having application-specific strategies of a parent JAVA application. In detail, the Abstract of '427 is as follows:

A system and [sic] method for efficiently providing access by a large number of clients to objects located at a large number of information servers. A non-bottleneck solution is provided to sharing load among servers by migrating or replicating objects over from highly loaded servers to less highly loaded servers. Objects that experience low loading are deleted to make room for more highly used objects and to permit make space for new objects. A naming service is provided to provide rapid access to a replica of a [sic] requested objects, while avoiding directing access requests to servers from which replicas of requested objects have been deleted. Hierarchical ordering of replication and naming functions permits a variety of particular access methods to be realized.

Taking the '427 Abstract sentence by sentence, sentence 1 refers to clients, objects, and servers for the objects. This does not indicate that any control module of the objects, or the servers for the objects, have application-specific strategies, or that any such strategies are provided to the platform via the control modules. This does not indicate that '427 teaches a method for starting a child application having application-specific strategies of a parent JAVA application.

Sentence 2 refers to load sharing among the servers, but does not indicate that the migration or replication of objects is based on application-specific strategies of a control module of a host application on any such server. Sentence 2 also does not indicate that the migration or replication starts a child application having application-specific strategies of a parent JAVA application.

Sentence 3 refers to deleting low loading objects, but this does not indicate that the deleting of those objects is pursuant to such an application-specific strategy, or relates to the starting of a child application having application-specific strategies of a parent JAVA application.

Sentence 4 refers to a naming service, and again, this does not indicate that control modules of an application have application-specific strategies provided to a platform, or that

the replica is a child application having application-specific strategies of a parent JAVA application.

Sentence 5 refers to hierarchical ordering of replication and naming functions. This does not indicate that control modules of the objects have application-specific strategies, nor that any of the replication or naming functions is part of a control module of an application, nor that the replication is of a child application having application-specific strategies of a parent JAVA application.

In review, the '427 Abstract does not provide any reason for changing the prior conclusion asserted in the prior Amendment and Response, that, even if combined, the combined references would not teach the claimed inventions.

Those prior remarks did not expressly refer to the cited '427 at C12, L15-30, cited against claims 3, 4, 15, 17, and 19 for the asserted teaching of application-specific policies including application-specific start policies (claim 3) or application-specific stop policies (claims 4 and 15) or a child control module including application-specific policies of a parent control module (claim 19), for example.

Reference is made to '427 at C12, L15-30. As to the cite of this C12 text to claim 3 (Action page 3), claim 3 defines the control module...includes application-specific policies, wherein the application-specific policies include application-specific start polices, and claim 4 defines the corresponding stop policies. It is respectfully submitted that the C12, L15-30 text describes a system protocol by which object migration occurs first via replica creation on the recipient node and then replica deletion from the source node (L27-30). L15-30 do not describe an application's control module including application-specific policies, wherein the application-specific policies include application-specific start polices. Rather, the cited C12

description is of coordination of replica creation and deletion with modifications of naming mappings of a name server (C12, L2-4). Moreover, the protocol for this coordination is shown as a system protocol by which (C12, L20-27) mapping on the name server occurs in the above sequence, e.g., modify mapping on the name server only after creation of the replica (L21-22). This is not a teaching of a claimed application-specific policy that includes application-specific start or stop policies because the naming protocol applies to all replications, i.e., on a system not application-specific basis.

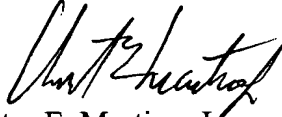
Further as to '427 at C12, L15-30, as to the cite of this C12 text to claims 15 and 17 (Action pages 6 and 7), claims 15 and 17 define a method for stopping a child application configured with a child control module having application-specific strategies in a JAVA environment, those policies including a application stop policy; and claim 19 defines the claim 15 method, wherein the child control module includes the application-specific policies of the parent control module. It is respectfully submitted that the C12, L15-30 described system protocol does not describe such a claimed child application with such claimed child control module. The described object migration via replica creation on the recipient node and then replica deletion from the source node does not teach a child application configured with a child control module having the claimed application-specific stop policy. As advanced above, the protocol for the described coordination is a system protocol by which (C12, L20-27) mapping on the name server occurs in the above sequence, i.e., modify mapping on the name server only after creation of the replica (L21-22), and a similar coordination occurs in re deletion by the name server of the mapping. Again, this is not a teaching of a claimed application-specific stop policy, because the naming protocol applies to all replications, i.e., on a system not application-specific basis.

In review, the '427 C12, L15-30 description does not provide any reason for changing the prior conclusion asserted in the prior Amendment and Response, that, even if combined, the combined references would not teach the claimed inventions.

In view of these supplemental remarks and the remarks in the prior Amendment and Response, all of which are filed before expiration of the period for response to the Office Action of 2/27/06, Applicants respectfully submit that all the pending claims 1, 3-13, and 15-19 are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested.

Respectfully submitted,

MARTINE PENILLA & GENCARELLA, L.L.P.



Chester E. Martine, Jr.
Reg. No. 19,711

Martine Penilla & Gencarella, LLP
710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
Telephone: (408) 774-6908
Customer Number 32291